Exp#4: LabVIEW with Arduino using LINX

What is LINX Toolkit?

LINX is an open source project by Digilent and is designed to make it easy to develop embedded applications using LabVIEW. LINX includes VIs for over 30 of the most common embedded sensors as well as hardware agnostic APIs for accessing peripherals like digital I/O, analog I/O, PWM, I2C, SPI, and UART. By using LINX toolkit, developers can acquire data from microcontroller and process it in the LabVIEW Graphical Programming environment.

Why LINX?

There are many reasons LabVIEW makes you more productive when using Arduino:

- Interact with your system through a graphical user interface.
- Streamline your design process with intuitive graphical programming.
- Improve your debugging experience with interactive tools.
- Leverage built in resources/functions for implementing simple to complex tasks.
- Open API allows for complete customization -- customize your programs to fit your application.

Introduction

In the previous, there was a toolkit called LIFA (LabView Interface For Arduino). The author of LIFA decided to expand the interface to more devices (and make other improvements). So, a few years ago, LIFA was replaced by LINX ("LabVIEW INterface for X"; where X stands for the various devices that it supports like Arduino, chipKIT, etc.). In addition, LINX is solved many problems that were in LIFA, so there were significant communication efficiency improvements because the protocol was redesigned so most of the functions that communicate with the device are much more efficient.

Step by Step Startup

Here is a step by step process to get up and running with Arduino and LabVIEW:

 Install the LINX. The LINX is available as a VI package through the LabVIEWTools Network. You must first install VIPM from the following link (http://jki.net/vipm/download). Then download the LINX package

VI 🔎 🖻 < 🗮	≕ \$		- 🗆 ×
	All 🗸 🕨	2019 ₆₄ 🗸 C	× Uninstall
Name	Version /\	Repository	Company
Xilinx ChipScopePro Debugging	2.1.0.11	NI LabVIEW Tools Network	B&A Engineering Inc. (BAI)
Digilent LINX (Control Arduino,	3.0.1.192	NI LabVIEW Tools Network	Digilent
<			>

Figure.1

After select the Digilent LINX press the install icon.

- Install NI-VISA Drivers. To LabVIEW, the arduino appears as a serial instrument device. To communicate with serial instruments in LabVIEW, you need to have the latest version of the NI-VISA driver. You can get the latest NI_VISA driver from "http://search.ni.com/nisearch/app/main/p/bot/no/ap/tech/lang/en/pg/1/sn/catnav:d u,n8:3.1637,ssnav:sup/".
- 3. Use the following steps to connect LabView with Arduino:
 - Go to Tools \rightarrow MakerHub \rightarrow LINX \rightarrow LINX Firmward Wizard.
 - Then select the following:

INA FIrmware Wizard	— — ×
NX rmware Wizard	
Device Family	
Arduino	HERE BERNART DINIZONE
Device Type	
Device Type Arduino Uno	
Device Type Arduino Uno Firmware Upload Method	



Then click Next and select Arduino port: ____ X 🜔 LINX Firmware Wizard _ LINX **Firmware Wizard** Select the COM port to use when uploading firmware to the Arduino Uno ASRL1: F ₽ COM4 Refresh Cancel Help Previous Next

Then click Next:

LINX Firmware Wizard	- 🗆 X
LINX Firmware Wizard	LabVIEW MakerHub
Firmware Version	
Upload Type	
Help	Previous Next Cancel

Figure. 4

After that LINX will start to make the connection. Once this step done, press finish and now you are ready to use Arduino with LabView.

Procedures

A. Control LED 13 in Arduino by LabView.



In this example, we will control LED 13 in Arduino by using LabView, so as shown in the figure above, we used a digital write box that has three inputs, the first input comes from init box, this input used to know that the initialization done successfully, the second input is a channel number, in other word, the digital port that we need to write on it, and the last input is a switch in order to control the led.

The front-end side will look like this:



B.Array example

In this example you will be use two binary arrays in as indicator and the other as control.

() o	Indicator Array
(÷) o	Control Array
	1111111

Figure.3 array example

- ☑ To add array: Front panel >> Modern >> Array, Matrix >> array.
- To make it a control binary array drags and drops a switch in it.
- To make it an indicator binary array drags and drops a switch in it.
- To re size the array graphically.

C. Read analog value (LDR)

Build the following LDR voltage divider circuit in to read its output voltage using arduino and print it out on the labview.



Figure.4 LDR voltage divider

Connect the output voltage of the LDR voltage divider circuit to analog bin 0 of Adriano. Then build the following example.



To add the "init, close, and read analog pin" block diagram >>LINX.

TODO: Lights detection security key (LDSK)

In this part You have to build a 12 bits light detection security key using a LDR with **Arduino and Labview**, where you will insert the password using the LDR if it detects a light it will be a one otherwise it will be zero.

- Use for loop to get the data from arduino and store them in binary array.
- [?] One bit every second.
- System needs a trigger to start reading. (you can use a Ok button with casestructure).
- If the password is correct, then a green light is on. Else a red one is on.

Lab Report:

1. Report as described in the Lab Policies.

2. Screen Shots of the "Block Diagram" and "Front Panel" windows where only these windows and their title bars and nothing else are visible. Points will be deducted if the components are not clearly visible in these windows or of any of the Windows screen components like the "Start Button" or "Task Bar" are visible.